Acta Cybernetica **17** (2006) 521–531.

On Armstrong Relations for Strong Dependencies

Vu Duc Thi* and Nguyen Hoang Son†

Abstract

The strong dependency has been introduced and axiomatized in [2], [3], [4], [5]. The aim of this paper is to investigate on Armstrong relations for strong dependencies. We give a necessary and sufficient condition for an abitrary relation to be Armstrong relation of a given strong scheme. We also give an effective algorithm finding a relation r such that r is Armstrong relation of a given strong scheme $G = (U, S)$ (i.e. $S_r = S^+$, where S_r is a full family of strong dependencies of r, and S^+ is a set of all strong dependencies that can be derived from S by the system of axioms). We estimate this algorithm. We show that the time complexity of this algorithm is polynomial in $|U|$ and $|S|$.

1 Introduction

Let us give some necessary definitions and results that are used in next section.

Definition 1. *Let U be a nonempty finite set of attributes, $r = \{h_1, \ldots, h_m\}$ a relation over U, and $A, B \subseteq U$. We say that B strongly depends on A in r (denote $A \xrightarrow[r]{s} B$) iff*

$$(\forall h_i, h_j \in r)((\exists a \in A)(h_i(a) = h_j(a) \Rightarrow (\forall b \in B)(h_i(b) = h_j(b))).$$

Let $S_r = \{(A, B) : A \xrightarrow[r]{s} B\}$. S_r is called a full family of strong dependencies of r. Where we write (A, B) or $A \to B$ for $A \xrightarrow[r]{s} B$ when r, s are clear from the context.

Definition 2. *A strong dependency (SD) over U is a statement of form $X \to Y$, where $X, Y \subseteq U$. The SD $X \to Y$ holds in a relation r if $A \xrightarrow[r]{s} B$. We also say that r satisfies the SD $A \to B$.*

Definition 3. *Let U be a set of attributes and $\mathcal{P}(U)$ its power set. Let $Y \subseteq \mathcal{P}(U) \times \mathcal{P}(U)$. We say that Y is an s – family over U iff for all $A, B, C, D \subseteq U$ and $a \in U$*

*Institute of Information Technology, Vietnamese Academy of Science and Technology, 18 Hoang Quoc Viet, Hanoi, Vietnam
†Department of Mathematics, College of Sciences, Hue University, Vietnam

(S1) $(\{a\},\{a\}) \in Y$,
(S2) $(A,B) \in Y, (B,C) \in Y, B \neq \emptyset \Rightarrow (A,C) \in Y$,
(S3) $(A,B) \in Y, C \subseteq A, D \subseteq B \Rightarrow (C,D) \in Y$,
(S4) $(A,B) \in Y, (C,D) \in Y \Rightarrow (A \cup C, B \cap D) \in Y$,
(S5) $(A,B) \in Y, (C,D) \in Y \Rightarrow (A \cap C, B \cup D) \in Y$.

It is easy to see that S_r is an $s-$ family over U.

It is known [4] that if Y is an $s-$ family over U, then there exists a relation r such that $Y = S_r$.

Definition 4. *A strong scheme G is a pair (U,S), where U is a finite set of attributes, and S a set of SDs over U.*

Let S^+ be a set of all SDs that can be derived from S by the rules in Definition 3.

It can be seen [4] that if $G = (U,S)$ is a strong scheme then there is a relation r over U such that $S_r = S^+$. Such a relation is called Armstrong relation of G.

Definition 5. *Let K be a Sperner-system over U. We define the set of antikeys of K, denoted by K^{-1}, as follows:*

$$K^{-1} = \{A \subset U : (B \in K) \Rightarrow (B \not\subseteq A) \text{ and } (A \subset C) \Rightarrow (\exists B \in K)(B \subseteq C)\}.$$

Definition 6. *The mapping $F : \mathcal{P}(U) \longrightarrow \mathcal{P}(U)$ is called a strong operation over U if for every $a, b \in U$ and $A \in \mathcal{P}(U)$ the following properties hold:*

(1) $a \in F(\{a\})$,

(2) $b \in F(\{a\})$ *implies* $F(\{b\}) \subseteq F(\{a\})$,

(3) $F(A) = \bigcap_{a \in A} F(\{a\})$.

Remark 7. It is clear that for arbitrary strong operation F

(1) $F(\emptyset) = U$,

(2) For all $A, B \in \mathcal{P}(U) : F(A \cup B) = F(A) \cap F(B)$,

(3) If $A \subseteq B$ then $F(B) \subseteq F(A)$.

It can be seen that the set $\{F(\{a\}) : a \in U\}$ determines the set $\{F(A) : A \in \mathcal{P}(U)\}$.

The following theorem shows that between $s-$ families and strong operations there exists a one - to - one correspondence

Theorem 8. [7] *Let S be a s – family over U. We define the mapping F_S as follows: $F_S(A) = \{a \in U : (A, \{a\}) \in S\}$. Then F_S is a strong operation over U. Conversely, if F is a strong operation over U then there is exactly one s – family S over U such that $F_S = F$, where $S = \{(A,B) : B \subseteq F(A)\}$.*

Definition 9. *Let $G = (U, S)$ be a strong scheme over U, $A \subseteq U$. We set*

$$A^+ = \{a \in U : A \to \{a\} \in S^+\}.$$

A^+ is called the closure of A over G.

It is clear that $A \to B \in S^+$ iff $B \subseteq A^+$.

Lemma 10. *Let $G = (U, S)$ be a strong scheme over U. Suppose that $A = \{a_1, \dots, a_k\}$ and $B = \{b_1, \dots, b_l\}$ are subsets of U. Then $A \to B \in S^+$ if and only if $\{a_i\} \to \{b_j\} \in S^+$ for every $i = 1, \dots, k; j = 1, \dots, l$.*

Proof. By rules (S3), (S4) and (S5), the lemma is obvious. □

Algorithm 11. [6] (Finding $\{a\}^+$)

Input: given a strong scheme $G = (U, S)$, where $S = \{A_i \to B_i : i = 1, \dots, m\}, a \in U$.

Output: compute $\{a\}^+$.

Method: we compute $\{a\}^+$ by induction.

Step 1. We set $X^{(0)} = \{a\}$.

Step i+1. If there is a SD $A_j \to B_j \in S$ so that $A_j \cap X^{(i)} \neq \emptyset$ and $B_j \not\subseteq X^{(i)}$ then we set

$$X^{(i+1)} = X^{(i)} \cup \Big(\bigcup_{A_j \cap X^{(i)} \neq \emptyset} B_j \Big).$$

In the converse case we set $\{a\}^+ = X^{(i)}$.

It is easy to see that there is a k such that $\{a\} = X^{(0)} \subseteq X^{(1)} \subseteq \cdots \subseteq X^{(k)} = X^{(k+1)} = \cdots$ and we set

$$\{a\}^+ = X^{(k)}.$$

Proposition 12. [6] *For each $a \in U$ Algorithm 11 computes $\{a\}^+$.*

It can be seen that the complexity of Algorithm 11 is polynomial time in the $|U|, |S|$.

Proposition 13. [6] *Let $G = (U, S)$ be a strong scheme over U, and $A \to B$ is a SD. Then there is a polynomial time algorithm deciding whether $A \to B \in S^+$.*

2 Armstrong Relation for Strong Dependency

It is known [8] that there is an algorithm that finds a set of all antikeys from a given Sperner-system.

Algorithm 14. [8]

Input: a Sperner-system $K = \{B_1, \dots, B_m\}$ over U.

Output: K^{-1}.

Method:

Step 1. We set $K_1 = \{U - \{a\} : a \in B_1\}$. It is clear that $K_1 = \{B_1\}^{-1}$.

Step q+1 ($q < m$). We suppose that $K_q = F_q \cup \{X_1, \ldots, X_{t_q}\}$, where $X_1, \ldots, X_{t_q}$ containing B_{q+1} and $F_q = \{A : A \in K_q, B_{q+1} \not\subseteq A\}$. For all i ($i = 1, \ldots, t_q$) we construct the antikeys of $\{B_{q+1}\}$ on X_i in an analogous way as K_1. Denote them by $A^i_1, \ldots, A^i_{r_i}$ $(i = 1, \ldots, t_q)$. Let

$$K_{q+1} = F_q \cup \{A^i_p : A \in F_q \Rightarrow A^i_p \not\subset A, 1 \leq i \leq t_q, 1 \leq p \leq r_i\}.$$

We set $K^{-1} = K_m$.

Theorem 15. [8] *For each* q $(1 \leq q \leq m), K_q = \{B_1, \ldots, B_q\}^{-1}$, *i.e.* $K_m = K^{-1}$.

It can be seen that K and K^{-1} are uniquely determined by one another and the determination of K^{-1} based on our algorithm does not depend on the order of $B_1, \ldots, B_m$. Denote $K_q = F_q \cup \{X_1, \ldots, X_{t_q}\}$ and let l_q $(1 \leq q \leq m-1)$ be the number of elements of K_q.

Proposition 16. [8] *The worst-case time complexity of our Algorithm 14 is*

$$\mathcal{O}(|U|^2 \sum_{q=1}^{m-1} t_q u_q),$$

where

$$u_q = \begin{cases} l_q - t_q & \text{if } l_q > t_q, \\ 1 & \text{if } l_q = t_q. \end{cases}$$

Note that $l_q \geq t_q$. Clearly, in each step of our algorithm K_q is a Sperner-system. In the cases for which $l_q \leq l_m (q = 1, ..., m-1)$, it is easy to see that the time complexity of our algorithm is not greater than $\mathcal{O}(|U|^2|K||K^{-1}|^2)$. Hence, in these cases Algorithm 14 finds K^{-1} in polynomial time in $|U|, |K|$ and $|K^{-1}|$. Obviously, if the number of elements of K is small, then Algorithm 14 is very effective. It only requires polynomial time in $|U|$.

Definition 17. *Let* $G = (U, S)$ *be a strong scheme over* U, *and* $a \in U$. *We set*

$$K_a = \{A \subseteq U : A \to \{a\} \in S^+, \nexists B : (B \to \{a\} \in S^+)(B \subset A)\}.$$

K_a *is called the family of minimal sets of the attribute* a.

Clearly, $\{a\} \in K_a, U \notin K_a$ and K_a is a Sperner-system over U.

Proposition 18. *Let* $G = (U, S)$ *be a strong scheme over* $U, a \in U$, K_a *is a family of minimal sets of* a *and* $n = |U|$. *Then*

(1) $K_a = \{\{b\} : b \in U, \{b\} \to \{a\} \in S^+\}$.

(2) $\forall A \in K_a : |A| = 1$.

(3) $|K_a| \le n$.

(4) $|K_a^{-1}| = 1$.

Proof. (1) We define the mapping $F_S : \mathcal{P}(U) \longrightarrow \mathcal{P}(U)$ as follows:

$$F_S(A) = \{a \in U : A \to \{a\} \in S^+\}.$$

By Theorem 8, it is clear that F_S is a strong operation over U. It is easy to see that $A^+ = F_S(A)$. Consequently, by Definition 6 we have

$$\begin{aligned} A^+ &= \bigcap_{a\in A} F_S(\{a\}) \\ &= \bigcap_{a\in A} \{b \in U : \{a\} \to \{b\} \in S^+\} \\ &= \bigcap_{a\in A} \{a\}^+. \end{aligned} \tag{1}$$

By (1) we obtain $A^+ \subseteq \{a\}^+ \quad \forall a \in A$. From this and the definition of K_a we immediately get

$$K_a = \{\{b\} : b \in U, \{b\} \to \{a\} \in S^+\}.$$

(2) It is obvious from (1).
(3) Because for each $A \in K_a : |A| = 1$, we can be seen that $|K_a| \le n$.
(4) By (2) and the definition of antikeys set, it is clear that $|K_a^{-1}| = 1$.
The proposition is proved. □

From this proposition we construct an algorithm finding a minimal set of the attribute a.

Algorithm 19. MSA

Input: a strong scheme $G = (U, S)$, and $a \in U$.

Output: $A \in K_a$.

Method:
MSA(G, a)
BEGIN
  Test:=true;
  WHILE test AND there is an attribute $b \in U$ such that

$$\{b\} \to \{a\} \in S^+$$

  DO BEGIN
    $A := \{b\}$;
    Test:=false
  END
  RETURN(A)
END.

Lemma 20. $A \in K_a$.

Proof. Because $\{a\} \in K_a$ and U is a finite set of attributes, the lemma is clear. □

The following lemma is obvious

Lemma 21. *The worst-case time complexity of MSA is* $\mathcal{O}(|U|^2|S|)$.

Remark 22. By Lemma 10 we have $A \to B \in S^+$ if and only if $\{a\} \to B \in S^+$ for every $a \in A$.

From this, we obtain the following lemma

Lemma 23. *Let* $G = (U, S)$ *be a strong scheme,* $a \in U, K_a$ *be a family of minimal sets of* $a, L \subseteq K_a, \{a\} \in L$. *Then* $L \subset K_a$ *if and only if there are* $C \in L, A \to B \in S^+$ *such that* $\forall E \in L \Rightarrow E \not\subseteq A \cup (C - B)$.

Proof. Suppose that $L \subset K_a$. Hence, there exists a $D \in K_a - L$. By $\{a\} \in L$ and the definition of K_a, we have

$$D \to \{a\} \in S^+ \tag{2}$$

and

$$a \notin D. \tag{3}$$

If for every SD $A \to B \in S$ implies $(A \cap D \neq \emptyset, B \subseteq D)$, or $A \cap D = \emptyset$, then $D^+ = D$. Therefore, by (3) we have $D \to \{a\} \notin S^+$. Which contradicts (2). Hence, there exists a SD $A \to B \in S$ such that $A \subseteq D$ and $B \not\subseteq D$. From this and Remark 22 we have a C such that $C \in L, A \subseteq D$ and $C - B \subseteq D$. Clearly, $A \cup (C - B) \subseteq D$. Consequently, we obtain $E \not\subseteq A \cup (C - B)$ for every $E \in L$.

Conversely, assume that there are $C \in L, A \to B \in S^+$ such that

$$E \not\subseteq A \cup (C - B) \tag{4}$$

for every $E \in L$. By the definition of L we have $A \cup (C - B) \to \{a\} \in S^+$. Because $\{a\} \in L$, there is a D such that $D \in K_a, a \notin D$ and $D \subseteq A \cup (C - B)$. From (4) we obtain $E \not\subseteq D$ for all $E \in L$, i.e. $D \in K_a - L$, or $L \subset K_a$.

The lemma is proved. □

From this lemma and MSA we construct the following algorithm by induction

Algorithm 24. FAMMSA

Input: a strong scheme $G = (U, S)$ and $a \in U$.

Output: K_a.

Method:

Step 1. Set $L(1) = E(1) = \{\{a\}\}$.

Step i+1. If there are C and $A \to B$ such that $C \in L(i), A \to B \in S, \forall E \in L(i) \Rightarrow E \not\subseteq A \cup (C - B)$, then by MSA construct an $E(i+1)$, where $E(i+1) \subseteq A \cup (C - B)$ and $E(i+1) \in K_a$. We set

$$L(i+1) = L(i) \cup E(i+1).$$

In the converse case we set $K_a = L(i)$.

By Lemma 23 there exists a natural number n such that $K_a = L(n)$.
The following lemma is obvious

Lemma 25. *The worst-case time complexity of FAMMSA is*

$$\mathcal{O}(|U|^2|S||K_a|(1 + |U||K_a|)).$$

By (3) in Proposition 18 we are easy to see that the time complexity of FAMMSA is polynomial in $|U|$ and $|S|$. Consequently, our algorithm is very effective.

It is obvious that if $S = \{\{a\} \to B_i : i = 1, \ldots, m\}$ or for each SD $A \to B \in S^+$ implies $a \notin B$, then $K_a = \{\{a\}\}$.

Let $G = (U, S)$ be a strong scheme over U. Set

$$\begin{aligned} MAX(S^+, a) = \{A \subseteq U : (A \to \{a\} \notin S^+) \\ \text{and } ((A \subset B) \Rightarrow (\exists D \subset B)(D \to \{a\} \in S^+)\}. \end{aligned}$$

It can be seen that

$$MAX(S^+, a) = K_a^{-1} \quad \forall a \in U. \tag{5}$$

Denote $MAX(S^+) = \bigcup_{a \in U} MAX(S^+, a)$.

Lemma 26. *If* $U - \cup MAX(S^+) \neq \emptyset$ *then*

$$\{c\} \to U \in S^+,$$

where for every $c \in U - \cup MAX(S^+)$.

Proof. Suppose that $c \in U - \cup MAX(S^+)$. Hence $c \notin \cup MAX(S^+)$. By (5) we have

$$\{c\} \notin K_a^{-1} \quad \forall a \in U.$$

According to Proposition 18 and the definition of set of antikeys we have

$$\{c\} \in K_a \quad \forall a \in U.$$

Consequently by (S5) in Definition 3 and the definition of K_a we immediately get

$$\{c\} \to U \in S^+.$$

The lemma is proved. □

Lemma 27. *For every* $b \in A, A \in K_a^{-1} : \{b\} \to \{c\} \notin S^+$, *where* $c \in U - A$.

Proof. Assume that there exists an $A \in K_a^{-1}$ and $b \in A$ such that $\{b\} \to \{c\} \in S^+$. Because $A \in K_a^{-1}$ and $c \in U - (A \cup \{a\})$, we have $\{c\} \in K_a$. Then by Proposition 18 we have

$$\{c\} \to \{a\} \in S^+, \quad a \in U.$$

Hence, by (S2) in Definition 3 we obtain

$$\{b\} \to \{a\} \in S^+.$$

Which contradicts the facts that $A \in K_a^{-1}$ and $b \in A$. Therefore, we have $\{b\} \to \{c\} \notin S^+ \forall b \in A, A \in K_a^{-1}$ and $c \in U - (A \cup \{a\})$.

The lemma is proved. □

Now we assume that $MAX(S^+) = \{A_1, \ldots, A_t\}$. Then we defined the mapping $Max : U \longrightarrow \mathcal{P}(U)$ as follows:

$$Max(a) = \begin{cases} \bigcap\limits_{a \in A_i} A_i & \text{if } \exists A_i \in MAX(S^+) : a \in A_i, \\ U & \text{otherwise.} \end{cases}$$

It is easy to see that $\forall a \in U : a \in Max(a)$, and hence $Max(a) \neq \emptyset$. On the other hand, we are easy to see that if $S = \{\{a_1\} \to U, \ldots, \{a_n\} \to U\}$ where $U = \{a_1, \ldots, a_n\}$ then

$$\forall a_i \in U : \quad Max(a_i) = U.$$

Lemma 28. *If* $Max(a) = \{a\} \cup A, A \neq \emptyset$ *and* $a \notin A$ *then* $\{a\} \to A \in S^+$.

Proof. First we suppose that there is $b \in A$ such that $\{a\} \to \{b\} \notin S^+$. By Proposition 18 we get $\{a\} \notin K_b$. Assume that $K_b^{-1} = \{\{a\} \cup B\}$. It is clear that $\{b\} \in K_b$. Hence $b \notin \cup K_b^{-1}$, i.e. $b \notin B$. It can be seen that if $B \neq \emptyset$ then $A \subseteq B$. Thus we obtain $b \in B$. This is a contradiction. Therefore, $B = \emptyset$ holds. By the definition of $Max(a)$ we obtain $Max(a) = \{a\}$. Which conflicts with the fact that $Max(a) = \{a\} \cup A, A \neq \emptyset$ and $a \notin A$. Consequently, we have

$$\{a\} \to \{b\} \in S^+ \quad \forall b \in A.$$

From this and according to (S5) in Definition 3 we immediately get

$$\{a\} \to A \in S^+.$$

The Lemma is proved. □

By Lemma 28 it is obvious that if $Max(a) = U$ then $\{a\} \to U \in S^+$.

The following theorem gives a necessary and sufficient condition for an arbitrary relation to be Armstrong relation of a strong scheme.

Theorem 29. *Let $G = (U, S)$ be a strong scheme, $r = \{h_1, \ldots, h_m\}$ a relation over U. Then a necessary and sufficient condition for r to be Armstrong relation of strong scheme G is*

$$\forall a \in U : \{a\}_r^+ = Max(a),$$

where $\{a\}_r^+ = \{b \in U : \{a\} \to \{b\} \in S_r\}$.

Proof. First we show that $\{a\}^+ = Max(a)$ for all $a \in U$. Denote $H = \{A_i : A_i \in MAX(S^+)$ and $a \in A_i\}$. It can be seen that if $H = \emptyset$ then according to Lemma 26 we get $\{a\} \to U \in S^+$.

Suppose that $H \neq \emptyset$. It is easy to see that if $H \subseteq MAX(S^+)$ holds then by Lemma 28 we have $\{a\} \to Max(a) \in S^+$.

By Lemma 27, it is obvious that for any M such that $M \supset Max(a)$ we have $\{a\} \to M \notin S^+$.

Consequently, according to the definition of $\{a\}^+$ we have

$$\forall a \in U : \{a\}^+ = Max(a). \tag{6}$$

Obviously, according to Theorem 8 we can see that $S_r = S^+$ iff for every $a \in U : \{a\}^+ = \{a\}_r^+$ holds. Hence, if $S_r = S^+$ holds then $\{a\}_r^+ = Max(a)$ for all $a \in U$.

Conversely, we suppose that $\{a\}_r^+ = Max(a)$ for all $a \in U$. Then by Theorem 8 and (6) we obtain $S_r = S^+$.

The theorem is proved. □

Now we construct an algorithm that from a given strong scheme G finds a relation r such that r is Armstrong relation of G.

Algorithm 30.

Input: a strong scheme $G = (U, S)$.

Output: a relation r such that $S_r = S^+$.

Method:

Step 1. By FAMMSA compute K_a for each $a \in U$.

Step 2. By Algorithm 14 we compute K_a^{-1} for each $a \in U$.

Step 3. Set

$$MAX(S^+) = \bigcup_{a \in U} K_a^{-1}.$$

Step 4. Denote elements of $MAX(S^+)$ by $A_1, \ldots, A_t$. We construct a relation $r = \{h_0, h_1, \ldots, h_t\}$ as follows

$$\text{for all } a \in U, \quad h_0(a) = 0, \quad \forall i = 1, \ldots, t$$

$$h_i(a) = \begin{cases} 0 & \text{if } a \in A_i, \\ i & \text{otherwise.} \end{cases}$$

By Theorem 29 we have r is an Armstrong relation of G, i.e. $S_r = S^+$.

The following example shows that for a given strong scheme G, Algorithm 30 can be applied to construct a relation r such that r is an Armstrong relation of G.

Example 31. A strong scheme $G = (U, S)$, where $U = \{a, b, c, d\}$ and $S = \{\{a, b\} \to \{c\}, \{b\} \to \{a, d\}, \{d\} \to \{b\}\}$.

Then we have
$K_a = \{\{a\}, \{b\}, \{d\}\}, K_b = \{\{b\}, \{d\}\}, K_c = \{\{a\}, \{b\}, \{c\}, \{d\}\}, K_d = \{\{b\}, \{d\}\}$.
$K_a^{-1} = \{\{c\}\}, K_b^{-1} = \{\{a, c\}\}, K_c^{-1} = \emptyset, K_d^{-1} = \{\{a, c\}\}$.
$MAX(S^+) = \{\{a, c\}, \{c\}\}$.
Consequently

$$r = \begin{array}{cccc} a & b & c & d \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 2 & 2 & 0 & 2 \end{array}$$

It is obvious that $S_r = S^+$.

Algorithm 32. [8]

Input: a Sperner-system $K_{a_i} = \{B_1, \ldots, B_{m_i}\}$ over U.

Output: $K_{a_i}^{-1}$.

Method:

Step 1. We set $K_{i_1} = \{U - \{a\} : a \in B_1\}$. It is clear that $K_{i_1} = \{B_1\}^{-1}$.

Step q+1 ($q < m_i$). We suppose that $K_{i_q} = F_{i_q} \cup \{X_1, \ldots, X_{t_{i_q}}\}$, where $X_1, \ldots, X_{t_{i_q}}$ containing B_{q+1} and $F_{i_q} = \{A : A \in K_{i_q}, B_{q+1} \not\subseteq A\}$. For all j ($j = 1, \ldots, t_{i_q}$) we construct the antikeys of $\{B_{q+1}\}$ on X_j in an analogous way as K_{i_1}. Denote them by $A_1^j, \ldots, A_{r_i}^j \quad (j = 1, \ldots, t_{i_q})$. Let

$$K_{i_{q+1}} = F_{i_q} \cup \{A_p^j : A \in F_{i_q} \Rightarrow A_p^j \not\subset A, 1 \le j \le t_{i_q}, 1 \le p \le r_j\}.$$

We set $K_{a_i}^{-1} = K_{i_m}$.

Denote $K_{i_q} = F_{i_q} \cup \{X_1, \ldots, X_{t_{i_q}}\}$ and $l_{i_q}(1 \le q \le m_i - 1)$ be the number of elements of K_{i_q}.

It is easy to see that the time complexity of Algorithm 30 is the time complexity of step 1 and step 2. By Proposition 16 and Lemma 25, the following proposition is clear.

Proposition 33. *The worst-case time complexity of Algorithm 30 is*

$$\mathcal{O}(n^2 \sum_{i=1}^{n} (\sum_{q=1}^{m_i - 1} t_{i_q} u_{i_q} + |S| m_i (1 + n m_i)))$$

where

$$U = \{a_1, \ldots, a_n\}, m_i = |K_{a_i}|,$$

$$u_{i_q} = \begin{cases} l_{i_q} - t_{i_q} & \text{if } l_{i_q} > t_{i_q}, \\ 1 & \text{if } l_{i_q} = t_{i_q}. \end{cases}$$

In the cases for which $l_{i_q} \leq l_{m_i}$ $(\forall i, \forall q : 1 \leq q \leq m_i)$, it is easy to see that the time complexity of our algorithm is

$$\mathcal{O}(n^2 \sum_{i=1}^{n} |K_{a_i}|(|S| + n|K_{a_i}||S| + |K_{a_i}^{-1}|^2)).$$

By (3) and (4) in Proposition 18 we are easy to see that the time complexity of Algorithm 30 is polynomial in $|U|$ and $|S|$. Consequently, our algorithm is very effective.

References

[1] Armstrong W. W., *Dependency structure of database relationship*, Information Processing 74, North-Holland Pub. Co. , (1974) 580-583.

[2] Czédli G., *Dependencies in the relational model of data (Hungarian)*, Alkalmaz Mat. Lapok **6** (1980), 131-143.

[3] Czédli G., *On dependencies in the relational model of data*, EIK **17** (1981), 103-112.

[4] Demetrovics J.,*Logical and structural investigation of relation datamodel*, MTA SZTAKI Tanulm-ányok **114** (1980), 1-97 (in Hungarian).

[5] Demetrovics J., Gyepesi G., *On the functional dependency and generalizations of it.* Acta Cybernetica Hungary **3** (1983), 295-305.

[6] Demetrovics J.,Thi V. D., *Armstrong relations, functional dependencies and strong dependencies*, Computers and Artificial Intelligence **14** (1995), 279-298.

[7] Thi V. D., *Strong dependencies and s-semilattices,* Acta Cybernetica **8** (1987), 195-202.

[8] Thi V. D., *Minimal keys and Antikeys*, Acta Cybernetica **7** (1986), 361-371.

[9] Thi V. D., Son N. H., *Some problems related to keys and the Boyce-Codd normal form*, Acta Cybernetica **16, 3** (2004), 473-483.

Received April, 2005